

CLAIMS

1
2
3 1. (Currently Amended) A method of synchronizing activation of
4 scheduled update data among a plurality of web servers, wherein each of the
5 plurality of web servers is coupled to a common data server, the method
6 comprising:

7 receiving a scheduled activation time from the common data server;

8 comparing a time associated with a clock in each web server of the plurality
9 of web servers to a time associated with a clock in the common data server;

10 adjusting the scheduled activation time on each web server of the plurality
11 of web servers by the time difference between the clock in each web server of the
12 plurality of web servers and the clock in the common data server;

13 prior to the scheduled activation time, receiving the scheduled update data
14 into staging caches in each web server of the plurality of web servers; and

15 at the scheduled activation time, activating the scheduled update data by
16 causing the scheduled update data from the staging caches within each web server
17 of the plurality of web servers to be accessible from an active cache within each
18 web server of the plurality of web servers.

19
20 2. (Canceled.)

21
22 3. (Previously presented) A method as recited in claim 1 wherein
23 each web server of the plurality of web servers contains a clock, and wherein the
24 clocks in each web server of the plurality of web servers are not synchronized with
25 one another.

1
2 4. (Previously Presented) A method as recited in claim 1 wherein
3 the causing the scheduled update data to be accessible from the active cache
4 comprises swapping an active data cache pointer with a staged data cache pointer.

5
6 5. (Original) A method as recited in claim 1 wherein no
7 communications are required between the individual web servers to synchronize
8 their data.

9
10 6. (Previously Presented) A method as recited in claim 1 wherein
11 the receiving scheduled update data into staging caches in each web server of the
12 plurality of web servers is performed asynchronously.

13
14 7. (Previously Presented) A method as recited in claim 1 further
15 comprising:

16 after the scheduled activation time, updating data caches in the common
17 data server.

18
19 8. (Original) A method as recited in claim 1 further comprising:
20 after the scheduled activation time, calculating a next scheduled activation
21 time.

1 9. (Previously Presented) A method as recited in claim 1 further
2 comprising:

3 after the scheduled activation time, updating data caches in the common
4 data server and calculating a next scheduled activation time, wherein the updating
5 and calculating are performed by the first web server to initiate a retrieval process
6 after the scheduled activation time.

7
8 10. (Previously Presented) A method as recited in claim 1 further
9 comprising:

10 if an additional web server is coupled to the common data server, then
11 causing the scheduled update data to be accessible from the active cache in the
12 common data server to an active cache in the additional web server.

13
14 11. (Previously Presented) A method as recited in claim 1 further
15 comprising:

16 if one of the plurality of web servers of the plurality of web servers is
17 initialized, then causing the scheduled update data to be accessible from the active
18 cache in the common data server to the active cache in the initialized web server.

19
20 12. (Original) A method as recited in claim 1 wherein the plurality of
21 web servers comprise a web farm.

22
23 13. (Original) A method as recited in claim 1 wherein the plurality of
24 web servers comprise a web farm, and wherein the plurality of web servers are
25 load balanced using a domain name service (DNS) round-robin technique.

1 14. (Original) One or more computer-readable memories containing
2 a computer program that is executable by a processor to perform the method
3 recited in claim 1.

4
5 15. (Previously Presented) A system comprising:
6 a web server of a plurality of web servers coupled to a common data server,
7 wherein the web server of the plurality of web servers comprises:

8 a staging cache;

9 an active data cache coupled to the staging cache; and

10 a clock having an associated time,

11 wherein the web server is configured to receive a scheduled activation time
12 from the common data server, and further configured to receive scheduled update
13 data from the common data server into the staging cache prior to the scheduled
14 activation time;

15 wherein the web server is configured to compare the time associated with
16 the clock in the web server to a time associated with a clock in the common data
17 server;

18 wherein the web server is further configured to adjust the scheduled
19 activation time on the web server by the time difference between the clock in the
20 web server and the clock in the common data server; and

21 wherein the web server is configured to cause the scheduled update data
22 from the staging cache to be accessible from the active data cache at the scheduled
23 activation time.

24
25 16. – 17. (Canceled.)

1 18. (Previously Presented) A system as recited in claim 15 wherein
2 the web server of the plurality of web servers contains a clock, and wherein the
3 clock in the web server of the plurality of web servers is not synchronized with
4 other web servers of the plurality of web servers.

5
6 19. (Original) A system as recited in claim 15 wherein the web server
7 is further configured to swap an active data cache pointer with a staged data cache
8 pointer.

9
10 20. (Previously Presented) A system as recited in claim 15 wherein
11 the web server of the plurality of web servers is configured to update data caches
12 in the common data server after the scheduled activation time.

13
14 21. (Previously Presented) A system as recited in claim 15 wherein
15 the web server of the plurality of web servers is configured to calculate a next
16 scheduled activation time after the scheduled activation time.

17
18 22. (Original) A system as recited in claim 15 wherein the plurality
19 of web servers comprise a web farm.
20
21
22
23
24
25

23. (Previously Presented) One or more computer-readable media having stored thereon a computer program that when executed performs a method comprising the following steps:

receiving a scheduled activation time from a common data server;
prior to the scheduled activation time, receiving scheduled update data into a staging cache in a web server;
comparing a time associated with a clock in the web server to a time associated with a clock in the common data server;
adjusting the scheduled activation time on the web server by the time difference between the clock in the web server and the clock in the common data server;
at the scheduled activation time, causing scheduled update data from the staging cache in the web server to be accessible from an active cache in the web server; and
after the scheduled activation time, updating data caches in the common data server and calculating a next scheduled activation time.

24. (Canceled.)

25. (Previously Presented) One or more computer-readable media as recited in claim 23 wherein the web server of a plurality of web servers contains a clock, and wherein the clock in the web server of the plurality of web servers is not synchronized with other web servers of the plurality of web servers.

1 26. (Previously Presented) One or more computer-readable media as
2 recited in claim 23 wherein updating data caches in the common data server and
3 calculating the next scheduled activation time are performed if another process has
4 not yet updated the data caches or calculated the next scheduled activation time
5 during a current data synchronization cycle.

6
7 27. (Previously Presented) One or more computer-readable media as
8 recited in claim 23 further comprising:

9 if the web server is initialized, then causing the scheduled update data to be
10 accessible from the active cache in the common data server to the active cache in
11 the initialized web server.

12
13 28. (Previously Presented) One or more computer-readable media as
14 recited in claim 23 wherein the causing the scheduled update data to be accessible
15 from the active cache comprises swapping an active data cache pointer with a
16 staged data cache pointer.

1 29. (Previously Presented) A method of synchronizing activation of
2 scheduled update data among each web server of a plurality of web servers,
3 wherein each web server of the plurality of web servers is coupled to a common
4 data server, the method comprising:

5 providing a scheduled activation time from the common data server to each
6 web server of the plurality of web servers;

7 communicating the scheduled update data into a staging cache in each web
8 server of the plurality of web servers prior to the scheduled activation time;

9 comparing a time recognized by each web server of the plurality of web
10 servers to a current time recognized by the common data server;

11 adjusting the scheduled activation time on each web server of the plurality
12 of web servers by the time difference between the time recognized by each web
13 server of the plurality of web servers and the current time recognized by the
14 common data server; and

15 causing the scheduled update data from the staging cache in each web
16 server of the plurality of the web servers to be accessible from an active cache in
17 each web server of the plurality of the web servers at the scheduled activation
18 time.

19
20 30. (Previously Presented) A method as recited in claim 29 wherein
21 the communicating scheduled update data into a staging cache is performed
22 asynchronously.
23
24
25

1 31. (Previously Presented) A method as recited in claim 29 wherein
2 the causing the scheduled update data to be accessible from the active cache
3 comprises swapping an active data cache pointer with a staged data cache pointer.
4

5 32. (Previously Presented) A method as recited in claim 29 wherein
6 no communication is required between each web server of the plurality of web
7 servers to synchronize their data.
8

9 33. (Original) One or more computer-readable memories containing
10 a computer program that is executable by a processor to perform the method
11 recited in claim 29.
12

13 34. – 35. (Canceled.)
14
15
16
17
18
19
20
21
22
23
24
25